



UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 862.C1926

First Named Inventor or Application Identifier

MINORU TORII

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO:

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☐ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)

2. ☒ Specification Total Pages 35

3. ☒ Drawing(s) (35 USC 113) Total Sheets 11

4. ☒ Oath or Declaration Total Pages 1

a. ☒ Newly executed (original or copy)

b. ☐ Unexecuted for information purposes

c. ☐ Copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 17 completed)
[Note Box 5 below]

d. ☐ **DELETION OF INVENTOR(S)**

Signed Statement attached deleting inventor(s)
named in the prior application, see 37 CFR
1.63(d)(2) and 1.33(b).

5. ☐ Incorporation By Reference (useable if Box 4c is checked)
The entire disclosure of the prior application, from which a copy of the
oath or declaration is supplied under Box 4c, is considered as being
part of the disclosure of the accompanying application and is hereby
incorporated by reference therein

6. ☐ Microfiche Computer Program (Appendix)

7. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)

a. ☐ Computer Readable Copy

b. ☐ Paper Copy (identical to computer copy)

c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & document(s))

9. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☐ Power of Attorney

10. ☐ English Translation Document (if applicable)

11. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations

12. ☐ Preliminary Amendment

13. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)

14. ☐ Small Entity Statement(s) ☐ Statement filed in prior application
Status still proper and desired

15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)

16. ☐ Other: _____

17. If a CONTINUING APPLICATION, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No. ____/____

18. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label

05514
(Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

NAME

Address

City

State

Zip Code

Country

Telephone

Fax

CLAIMS	(1) FOR	(2) NUMBER FILED	(3) NUMBER EXTRA	(4) RATE	(5) CALCULATIONS
TOTAL CLAIMS (37 CFR 1.16(c))		13-20 =	0	X \$ 18.00 =	\$0.00
INDEPENDENT CLAIMS (37 CFR 1.16(b))		3-3 =	0	X \$ 78.00 =	\$0.00
MULTIPLE DEPENDENT CLAIMS (if applicable) (37 CFR 1.16(d))				\$260.00 =	\$0.00
				BASIC FEE (37 CFR 1.16(e))	\$690.00
				Total of above Calculations =	\$690.00
				Reduction by 50% for filing by small entity (Note 37 CFR 1.9, 1.27, 1.28).	
				TOTAL =	\$690.00

19. Small entity status

- a. ☐ A Small entity statement is enclosed
- b. ☐ A small entity statement was filed in the prior nonprovisional application and such status is still proper and desired.
- c. ☐ Is no longer claimed.

20. ☒ A check in the amount of \$ 690.00 to cover the filing fee is enclosed.

21. ☒ A check in the amount of \$ 40.00 to cover the recordal fee is enclosed.

22. The Commissioner is hereby authorized to credit overpayments or charge the following fees to Deposit Account No. 06-1205:

- a. ☒ Fees required under 37 CFR 1.16.
- b. ☒ Fees required under 37 CFR 1.17.
- c. ☐ Fees required under 37 CFR 1.18.

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT REQUIRED

NAME Leonard P. Diana (Reg. No. 29,296)

SIGNATURE

DATE

June 13, 2000

TITLE OF THE INVENTION
NETWORK DEVICE MANAGING APPARATUS AND METHOD
AND STORAGE MEDIUM

5 FIELD OF THE INVENTION

 The present invention relates to a computer
network and, more particularly, to a managing apparatus
and method of managing devices connected to a network,
and a storage medium storing a program for causing a
10 computer to execute the managing method.

BACKGROUND OF THE INVENTION

 Fig. 1 shows an arrangement in which a network
board (NB) 101 for connecting a printer to a network is
15 connected to a printer 102 having an open architecture.
The NB 101 is connected to a local area network (LAN)
100 via a LAN interface such as the Ethernet Interface
10Base-2 having a coaxial connector or the 10Base-T
having RJ-45.

20 A plurality of personal computers (PCs) such as
PCs 103 and 104 are also connected to the LAN 100.
These PCs can communicate with the NB 101 under the
control of a network operating system. One of these
PCs, e.g., the PC 103, can be used for network
25 management. A local printer, such as a printer 105
connected to the PC 104, can be connected to a PC.

Furthermore, a file server 106 is connected to the LAN 100 to manage access to files stored in a network disk 107 having a large capacity (e.g., 10 gigabytes).

5 A print server 108 causes printers, such as connected printers 109 and the printer 105 in a remote place, to perform printing. Other peripheral devices (not shown) can also be connected to this LAN 100.

10 In addition, a WWW server 150 is connected to the LAN 100 so that an HTML (Hyper Text Markup Language) document created by network management software installed on the WWW server 150 can be displayed by using a WWW browser installed on the PC 103, or printer settings made on the WWW browser on the PC 103 can be
15 transmitted to a specific printer through the network management software on the WWW server 150.

More specifically, in the network shown in Fig. 1, network software such as Novell or Unix software can be used to efficiently perform communications between
20 various network members. It is possible to use any network software, e.g., the NetWare (trademark of Novell Corp.; this will be omitted hereinafter) of Novell Corp. A detailed explanation of this software package is made in on-line documentation of the NetWare
25 package. This can be purchased together with the NetWare package from Novell Corp.

In brief, the file server 106 functions as a file manager for receiving, storing, queuing, caching, and transmitting data files between LAN members. For example, data files formed by the PCs 103 and 104 are transmitted to the file server 106. The file server 106 arranges these data files in order and, in accordance with a command from the print server 108, transmits the arranged data files to one of the printers 109.

Each of the PCs 103 and 104 is a common PC capable of forming a data file, transmitting the formed data file to the LAN 100, receiving a file from the LAN 100, and displaying and/or processing such files. Although personal computers are shown in Fig. 1, other computers suited to executing the network software can also be included. For example, when UNIX software is used, UNIX workstations can be included. These workstations can be used together with the PCs shown in Fig. 1 under appropriate conditions.

Commonly, a LAN such as the LAN 100 provides services to a relatively local user group, e.g., a user group on one floor, or across a plurality of consecutive floors, in one building. For example, when a certain user is away from other users, such as when a user exists in another building or in another prefecture, a wide area network (WAN) can be

constructed. The WAN is basically a group formed by connecting a plurality of LANs by high-speed digital lines such as an integrated services digital network (ISDN). Accordingly, as shown in Fig. 1, the LAN 100, 5 a LAN 110, and a LAN 120 form a WAN as they are connected via a modem/transponder 130 and backbone 140.

Each LAN includes dedicated PCs and may include a file server and a print server, as needed. As shown in Fig. 1, therefore, the LAN 110 includes PCs 111 and 112, 10 a file server 113, a network disk 114, a print server 115, and printers 116. In contrast, the LAN 120 includes only PCs 121 and 122. Devices connected to any of these LANs 100, 110, and 120 can access functions of devices of the other LANs via the WAN 15 connections.

As a method of managing devices on networks constructing such a large-scale network system, a large number of standard organizations have made several attempts. An International Standardization 20 Organization (ISO) has provided a versatile reference framework called an Open System Interconnection (OSI) model. The OSI model of a network management protocol is called a Common Management Information Protocol (CMIP). This CMIP is a common network management 25 protocol in Europe.

the user. To extract the data, a corresponding process is called in some cases.

Each agent holds data concerning its own state in the form of a database. This database is called a
5 Management Information Base (MIB). The MIB has a tree data structure, and all nodes are uniquely numbered. This node identifier is called an object identifier.

This MIB structure is called a Structure of Management Information (SMI) and defined by RFC1155
10 Structure and Identification of Management Information for TCP/IP-based Internets.

The SNMP will be briefly described below. A PC (to be referred to as a manager hereinafter), in which network management utility software is operating, and a
15 management objective network device (to be referred to as an agent hereinafter), in which an SNMP agent is operating, communicate with each other by using the SNMP. This SNMP has five types of commands written as Get-request, Get-next-request, Get-response,
20 Set-request, and Trap.

Get-request and Get-next-request are commands which are sent from the manager to the agent to acquire the value of an MIB object of the agent. Upon receiving these commands, the agent sends the
25 Get-response command to the manager to inform the manager of the MIB value.

Set-request is a command which is sent from the manager to the agent to set the value of the MIB object of the agent. Upon receiving this command, the agent sends the Get-response command to the manager to inform
5 the manager of the set result.

Trap is a command which is sent from the agent to the manager to inform the manager of a change in the state of the agent.

In a well-known system, an SNMP agent operates on
10 the network board (NB) 101 connected to the PC and printer 102, and network management software serving as an SNMP manager operates on the PC. With the recent proliferation of the Internet, a system has been developed, which makes network management software
15 operate on a server, instead of making dedicated network management software operate on each client PC, and also using the WEB as a user interface.

The operation of a general WWW system and the operation of an SNMP management program based on the
20 WWW system will be briefly described next with reference to Fig. 2.

A WWW server program 1051 operates on a PC 150. Many WWW page data described by using HTML are stored in a disk in the PC 150.

25 A WWW browser program 1031 operating on a PC 103 requests the WWW server program 1051 operating on the

PC 150 to send a page designated by a user in order to display the page.

In response to the request from the WWW browser program 1031, the WWW server program 1051 returns the designated page data. The WWW browser program 1031 analyzes the acquired page data and displays the page in accordance with the description.

If a request having passed through the CGI (Common Gateway Interface) is contained in the page acquisition request from the WWW browser program 1031, the WWW server program 1051 activates an external script or program by a predetermined method based on the CGI, receives page data for a response to the request from the WWW browser program 1031, and returns it to the WWW browser program 1031.

A case wherein an external program activated by the CGI is a network management program as in the present invention will be described next.

A network management program 1052 activated by the CGI under the control of the WWW server program 1051 acquires management data from a device, e.g., the printer 102, which is connected to the network by using the SNMP. The network management program 1052 creates a page described in HTML on the basis of the acquired management data, and returns the page to the WWW server program 1051.

An application for managing devices by the
SNMP/MIB uses a WWW browser program. Such application
uses a template file constituted by unique keywords and
an HTML description to implement a user interface on
5 the WWW browser program.

According to the prior art described above,
network management program is activated every time a
request is received from a user (Web browser). Assume
that the server is not holding link destination
10 information of a window currently displayed by the
browser, which is likely to be requested from the user.
In this case, if information between the links is
requested by the user, the server acquires the
information upon communicating with the corresponding
15 device again. That is, a long processing time is
required to display the information.

In addition, even if the server is holding link
destination information, since information in the
database is updated at a timing different from that of
20 a request from the user, the link destination
information loses its real-time property.

SUMMARY OF THE INVENTION

In order to achieve the above object, a network
25 device managing apparatus and method according to the

present invention have at least the following arrangements.

There is provided a network device managing apparatus using an SNMP protocol, comprising generating
5 means for specifying a device on a network as a management target and generating a command for acquiring management information for the device and setting the information in a memory, setting means for setting, on the basis of the command, management
10 information of the device which is acquired through the network in the memory, and output means for outputting/displaying the set or acquired management information of the device in a predetermined form.

In the network device managing apparatus, the
15 management information is preferably information in an MIB form.

In the network device managing apparatus, the command preferably contains an HTML format for defining the predetermined form, and a management information
20 item of the device.

In the network device managing apparatus, the output means preferably displays the set or acquired result in an HTML format.

In the network device managing apparatus, if
25 there is an URL linked to the management information,

the setting means preferably further sets or acquires linked management information.

In the network device managing apparatus, the output means preferably displays a result of management
5 information set or acquired in accordance with the URL.

In addition, there is provided a network device managing method using an SNMP protocol, comprising the generating step of specifying a device on a network as a management target and generating a command for
10 acquiring management information for the device and setting the information in a memory, the setting step of setting, on the basis of the command, management information of the device which is acquired through the network in the memory, and the output step of
15 outputting/displaying the set or acquired management information of the device in a predetermined form.

In the network device managing method, the management information is preferably information in an MIB form.

20 In the network device managing method, the command preferably contains an HTML format for defining the predetermined form, and a management information item of the device.

In the network device managing method, the output
25 step preferably comprises displaying the set or acquired result in an HTML format.

In the network device managing method, if there is an URL linked to the management information, the setting step preferably further comprises setting or acquiring linked management information.

- 5 In the network device managing method, the output step preferably comprises displaying a result of management information set or acquired in accordance with the URL.

- Furthermore, there is provided a
- 10 computer-readable storage medium storing a program for managing network devices using an SNMP protocol, the program comprising a code for the generating step of specifying a device on a network as a management target and generating a command for acquiring management
- 15 information for the device and setting the information in a memory, a code for the setting step of setting, on the basis of the command, management information of the device which is acquired through the network in the memory, and a code for the output step of
- 20 outputting/displaying the set or acquired management information of the device in a predetermined form.

- Other features and advantages of the present invention will be apparent from the following description taken in conjunction with the accompanying
- 25 drawings, in which like reference characters designate the same or similar parts throughout the figures

thereof.

BRIEF DESCRIPTION OF THE DRAWINGS

The accompanying drawings, which are incorporated
5 in and constitute a part of the specification,
illustrate embodiments of the invention and, together
with the description, serve to explain the principles
of the invention.

Fig. 1 is a block diagram showing an arrangement
10 in which a printer having a network board is connected
to a network;

Fig. 2 is a view for explaining an outline of the
operation of an SNMP management program;

Fig. 3 is a block diagram showing the arrangement
15 of a PC on which network management software operates;

Fig. 4 is a block diagram showing the module
configuration of the network management software
according to the present invention;

Fig. 5 is a view showing the arrangement of a
20 template file according to the present invention;

Fig. 6 is a view showing a device list window;

Fig. 7 is a view showing an example of a display
window for showing the details of a device;

Fig. 8 is a block diagram showing the file
25 configuration of the network management software;

Fig. 9 is a flow chart for explaining a procedure for acquiring management information of a network device and displaying the result;

Fig. 10 is a flow chart for explaining a
5 procedure for a linked URL; and

Fig. 11 is a flow chart for explaining a procedure for displaying URL information.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

10 Preferred embodiments of the present invention will be now be described in detail in accordance with the accompanying drawings.

Network management software (network management program) of the present invention is constituted by a
15 PC 103 on which a Web browser like the one shown in Fig. 1 can operate, a WWW server 150, and a printer 102 connected to a network board 101 having the function of the SNMP/MIB.

Fig. 3 shows the arrangement of a PC on which
20 this network management software can operate. Referring to Fig. 3, the WWW server 150 is a PC on which the network management software operates, and equivalent to the WWW server 150 shown in Fig. 1. The PC 150 includes a CPU 301 which executes a network
25 management program stored in a ROM 302 or in a hard disk (HD) 311 or supplied from a floppy disk (FD) 312,

and comprehensively controls individual devices connected to a system bus 304.

A RAM 303 functions as a main memory and work area for the CPU 301. A keyboard controller (KBC) 305 controls input instructions from a keyboard (KB) 309, a pointing device (not shown), and the like. A CRT controller (CRTC) 306 controls the display of a CRT display (CRT) 310. A disk controller (DKC) 307 controls access to the hard disk (HD) 311 and floppy disk (FD) 312 storing a boot program, various applications, edit files, user files, the network management program, and the like. A network interface card (NIC) 308 bidirectionally exchanges data with agents or network devices via a LAN 100.

The hard disk (HD) 311 stores a program of the network management software according to the present invention, which is the main part of operation in the entire explanation to be described later. In the entire explanation described later, the main hardware part of execution is the CPU 301 unless otherwise specified. On the other hand, the main software part of control is the network management software stored in the hard disk (HD) 311. In this embodiment, Windows NT (Microsoft Corp.) and IIS (Internet Information Server) are assumed as an OS and WWW server, respectively.

However, an OS and WWW server are not particularly limited.

The network management program according to the present invention can also be supplied as it is stored in a storage medium such as a floppy disk or a CD-ROM. In this case, the program is read out from the storage medium by the floppy disk controller (FD) 312 shown in Fig. 3 or a CD-ROM driver (not shown) and installed in the hard disk (HD) 311.

Fig. 4 shows the module configuration of the network management software according to the present invention.

Network management software 1062 according to the present invention is stored in the hard disk 311 shown in Fig. 3 and executed by the CPU 301. During the execution, the CPU 301 uses the RAM 303 as a work area.

Referring to Fig. 4, the network management software 1062 is activated by a WWW server program 1061 to exchange CGI parameters and HTML documents through a CGI interface 402.

A system control module 403 registers CGI parameters in a parameter module 404 (to be described later), and then transfers control to a system module 405, device list module 407, or device detail module 409 (to be described later) in accordance with a command parameter in the CGI parameters. If there is

an error in the CGI parameters, the system control module 403 may create an HTML document indicating the presence of the error in the CGI parameters through a template module 412.

5 The parameter module 404 stores/manages the CGI parameters, registered by the system control module 403, in a tabular form. Other modules can acquire desired parameters from the CGI interface 402, as needed.

10 The system module 405 controls display/setting of system parameters (e.g., automatic update interval for HTML documents) that define the operation of the network management software 1062, and creates associated HTML documents. The system module 405 acquires a command parameter from the parameter module 15 404. If the contents of the command parameter indicate a system parameter display request, the system module 405 reads out necessary information from a system setting file 406, and creates an HTML document for system parameter display through the template module 20 412. If the contents of the acquired command parameter indicate a system parameter setting request, the system module 405 writes the informed system parameter in the system setting file 406, and creates an HTML document to be displayed after setting through the template 25 module 412. Although not shown, system parameters stored in the system setting file 406 can be read out

by the respective modules constituting the network .
management software 1062, as needed.

The device list module 407 creates an HTML
document indicating a list of devices (device list)
5 searched out by a device search module 408 (to be
described later). The device list module 407 also
controls processing for a device list display option or
the like.

The device search module 408 searches for a
10 device connected to the network.

The device detail module 409 performs control to
display/set detailed information about a specific
device designated by a CGI parameter, and also creates
an associated HTML document. The device detail module
15 409 uses a device native module 410 (to be described
later) corresponding to the designated device to
acquire/set detailed information about the designated
device.

The device native module 410 is prepared for each
20 device (a printer, network interface, or the like) to
be managed by the network management software. In
display operation, the device native module 410
acquires necessary information from a device, and sets
the acquired information in the template module 412.
25 In setting operation, the device native module 410
converts a set value informed by a CGI parameter into a

value that can be interpreted by a device, and transmits the value to the device.

A protocol module 411 performs control on various protocols required for the network management software to communicate with devices, e.g., handling of the MIB (Management Information Base), transmission/reception of SNMP (Simple Network Management Protocol) packets, and control on a transport protocol.

A template module 412 creates an HTML document as an output result of the network management software on the basis of a template file 413 stored in the hard disk 311 in Fig. 3.

The template module 412 opens a template file designated by a CGI parameter, the system control module 403, the system module 405, the device list module 407, or the device detail module 409, and analyzes the contents of the template file. The template module 412 then creates an HTML document by replacing the template variables contained in the template file with values set by the system control module 403, system module 405, device list module 407, device detail module 409, or device native module 410, and transmits the document to a WWW server program via the CGI interface 402. The values of the template variables used to create the HTML document or the created HTML document file can be stored as a cache

file 414 in the hard disk 311 in Fig. 3 to shorten the processing time required to create the second and subsequent HTML documents on the basis of the same template file.

5 Fig. 5 shows the format of a template file according to the present invention.

 The contents of a template file used in network management software 1062, which are shown in Fig. 5, are described between the <<TEMPLATE>> tag and the
10 <</TEMPLATE>> tag. The description between these tags is constituted by two blocks, the HEAD block described between <<HEAD>> tag and the <</HEAD>> tag and the BODY block described between the <<BODY>> tag and the <</BODY>> tag.

15 In the HEAD block, the <<VARIABLE>> tags in which pieces of information to be acquired from a device are described as variables are mainly described. The BODY block is constituted by an output HTML description, the <<EMBED>> tag in which values acquired from the device
20 are embedded, and the like.

 As is obvious from Fig. 5, network management software 1062 may acquire a product name indicated by WNTVAR_DCV_PRODUCT by parsing the HEAD block. Upon acquiring this information, network management software
25 1062 acquires information from the device by the SNMP/MIB. Thereafter, network management software 1062

replaces the <<EMBED>> tag having the variable "WNTVAR_DCV_PRODUCT" with the information acquired from the device by parsing the BODY block. By performing parsing operation like that described above with
 5 respect to all the variables, an HTML file to be output can be obtained.

The following is a list of tags used in a template file and their functions.

	Tag	Function
10	TEMPLATE	template description
	HEAD	header description
	BODY	body description
	LINK	associated template description
	VARIABLE	declaration of template variable
15	INCLUDE	inclusion of template file
	SET	setting of value of variable
	EMBED	embedding of variable value
	ISVALID	evaluation of validity of variable
	value	
20	EVAL	comparison between variable values
	LOOP	repetitive description
	COMMENT	comment description
	LINKURL	description of URL to be linked

Fig. 6 shows a window called a device list. When
 25 the user designates a URL indicating this window, network management software 1062 is activated through

the CGI. Network management software 1062 searches for devices connected to the network, and displays, on the browser, the following information about the devices detected by the search:

5 DEVICE TYPE
 DEVICE NAME
 PRODUCT NAME
 PRODUCT NAME OF NETWORK BOARD
 NETWORK ADDRESS

10 STATE OF DEVICE

 The type of device indicates whether the device is a commonly-used printer or a composite machine having a copy function as well, and is displayed as an icon. A device name is a name given to each device by
15 the user. When the user clicks this device name, network management software 1062 is activated again to display the detailed information about the device. This operation will be described in detail later. The state of the device is indicated by changing the icon
20 in accordance with the importance of a current error.

 When the user clicks a device name in the device list, an IP address is transferred to network management software 1062, and detailed information about the clicked device is acquired on the basis of
25 this information. The detailed information is then displayed on the browser. The display information

includes a state, equipment information, device
information, network board information, and protocol
information. These pieces of information are displayed
in a plurality of windows. Fig. 7 is an example of a
5 display window. Fig. 7 shows the details of a device.

Fig. 8 shows the file configuration of network
management software 1062. Each double-frame box
indicates a directory; and each single-frame, a file.
Reference numeral 901 denotes a root directory of
10 network management software 1062 as a CGI program. The
following directories are subordinate to this root
directory:

Document
Images
15 Template

In addition, WNS.exe as an execution file and various
HTML files are present.

In a Document directory 902, information acquired
from a device is temporarily stored as a cache file.
20 In an Images directory 903, various image files used
for the display of information are stored. In a
directory 904, a template file is stored. Three types
of directories are subordinate to the directory 904.

In a sys directory 908, a template file for the
25 display of information independent of devices, e.g., a
device list and error relations, is stored. In product

directories 909, template files associated with
information unique to products are stored. The number
of product directories is equal to the number of
product types. Information stored in each of these
5 directories includes a state, equipment information,
and device information shown in Fig. 7. In NIC
directories 910, information unique to network boards
is stored. Information stored in these NIC directories
equal in number to repairs to the network boards
10 includes network information and protocol information.

If the same template file can be created for a
plurality of products or network boards, directories
need not always be prepared in a number equal to the
number of types of products or network boards.

15 Fig. 9 is a flow chart showing a procedure for
making network management software 1062, activated on
the PC 150, acquire MIB information of a device and
display the information on the browser on the PC 103 in
a case wherein network management software 1062 on the
20 PC 150 on which the WWW server operates is activated
from the WWW browser on the PC 103 in Fig. 1 to manage
the printer 102 (in which the SNMP agent is installed).

In step S101 in the flow chart of Fig. 9, network
management software 1062 is invoked by sending, over
25 the Get command, a command identifier for indicating
whether to acquire or set device information from the

Web browser and a template identifier for identifying a template.

In step S102, network management software 1062 is activated through the CGI. In step S103, activated
5 network management software 1062 opens the template designated by the command identifier and template identifier sent over the Get command in step S101, and parses a list of information to be acquired from the device from the HEAD portion of the template file. As
10 shown in Fig. 5, the template file is constituted by two structures, namely the HEAD portion between <<HEAD>> and <</HEAD>> and the BODY portion between <<BODY>> and <</BODY>>. In the <<HEAD>> portion, useful information to be recognized by network
15 management software 1062 in advance is stored. For example, a list of variables representing information to be acquired from the device indicated by <<VARIABLE NAME =...>> is defined.

The <<BODY>> portion is made up of a description
20 in an HTML form which is to be displayed on the Web browser and template variables replaced with the information acquired from the device.

In step S104, MIB information is actually acquired from the device in accordance with the list of
25 variables acquired in step S103, i.e., the list of variables indicated by the <<VARIABLE NAME=...>> tags.

In step S105, the <<BODY>> portion is parsed by using the MIB information acquired in the step S104. More specifically, an HTML file to be output is created by replacing the template variable <<EMBED...>> with the value acquired from the device. When the <<BODY>> portion is parsed, linked URL information indicated by the <<LINKURL>> tag is stored. For example, the menu on the page showing the details of the device in Fig. 7 is linked to URLs indicating error information, network information, and protocol information. When the user presses this button, network management software is activated again to acquire error information, network information, or protocol information from the device and display the information on the Web browser.

As the URL indicating the network information, <<LINKURL=WNS.exe?cmd=devget&addr=192.168.16.132&tmpl=network>> is described in the template file. Information is therefore acquired from the device by using a template file named as network.wtf.

In step S106, the transmission file created in step S105 is output to STDOUT. With this operation, the information is displayed on the Web browser on the PC 103.

In step S107, it is checked whether there is any linked URL and the <<LINKURL>> tag is parsed. If there is a linked URL, the flow advances to step S108 to

process the linked URL. This processing will be described in detail with reference to the flow chart of Fig. 10. If there is no linked URL, the processing is terminated.

- 5 A method of acquiring information from a device on the basis of linked URL information will be described with reference to the flow chart of Fig. 10.

In step S201 in the flow chart of Fig. 10, WNS.exe is activated in another process on the basis of
10 the information acquired by parsing the <<LINKURL>> tag in step S105 in Fig. 9. For example, since the network information indicated by the device detail menu in Fig. 7 is described as
<<LINKURL=WNS.exe?cmd=devget&addr=192.168.132&tmpl=netw
15 ork>>, MIB information is acquired from the device whose IP address is 192.168.132 by using the network.wtf template file.

In step S202, a list of information to be acquired from the device is obtained by parsing the
20 template file designated by the <<LINKURL=...>> tag. As in step S103 of Fig. 9, this information is defined by a list of variables representing information to be acquired from the device indicated by <<VARIABLE
NAME=...>>.

- 25 In step S203, MIB information is acquired from the device on the basis of the list of information to

be acquired from the device which is obtained in step S202.

In step S204, the template variable <<EMBED...>> is replaced with the value acquired from the device by using the MIB information acquired in step S203, thereby creating an HTML file to be output. This file is stored as cache data in the Document directory in Fig. 8 without outputting to STDOUT.

Operation to be performed when the user is to be display linked URL information will be described with reference to the flow chart of Fig. 11.

In step S301 in the flow chart of Fig. 11, the user clicks the linked URL.

In step S302, the Web browser sends, over the Get command, a command identifier for indicating whether to acquire or set device information and a template identifier for identifying a template to be used, thereby invoking network management software 1062.

In step S303, network management software 1062 is activated through the CGI.

In step S304, cache data using the flow chart of Fig. 10 is output to STDOUT.

In step S305, the cache data is output to the Web browser, and the information of the linked URL is displayed.

The information of the URL associated with the linked device is acquired by another process that has activated network management software 1062, and designated variables are replaced with acquired values.

5 The resultant data is stored as cache data. This makes it possible to provide the cache data to the user without communicating with a device in response to every request from the user.

The information acquired in this manner is based
10 on the Get command sent from the user, the latest data on the network can be provided in real time.

Note that the present invention may be applied to either a system constituted by a plurality of devices (e.g., a host computer, an interface device, a reader, a
15 printer, and the like), or an apparatus consisting of a single device (e.g., a copying machine, a facsimile apparatus, or the like).

The objects of the present invention are also achieved by supplying a storage medium (or a recording
20 medium), which records a program code of a software program that can realize the functions of the above-mentioned embodiments to the system or apparatus, and reading out and executing the program code stored in the storage medium by a computer (or a CPU or MPU) of
25 the system or apparatus. In this case, the program code itself read out from the storage medium realizes the

functions of the above-mentioned embodiments, and the storage medium which stores the program code constitutes the present invention. The functions of the above-mentioned embodiments may be realized not only by
5 executing the readout program code by the computer but also by some or all of actual processing operations executed by an OS (operating system) running on the computer on the basis of an instruction of the program code.

- 10 Furthermore, the functions of the above-mentioned embodiments may be realized by some or all of actual processing operations executed by a CPU or the like arranged in a function extension board or a function extension unit, which is inserted in or connected to the
15 computer, after the program code read out from the storage medium is written in a memory of the extension board or unit.

When the present invention is applied to the above storage medium, program codes corresponding to the flow
20 charts (shown in Figs. 9, 10 and/or 11) described above are stored.

As has been described above, according to the network device managing apparatus and method of the present invention, the information of a linked URL is
25 acquired by another process and stored as cache data. This makes it possible to quickly provide device

information to a user without communicating a device in response to every request from the user.

In addition, since the linked information is obtained immediately before a user issues a request to
5 acquire link destination information, the information can be provided as information with an excellent real-time property to the user.

As many apparently widely different embodiments of the present invention can be made without departing from
10 the spirit and scope thereof, it is to be understood that the invention is not limited to the specific embodiments thereof except as defined in the claims.

WHAT IS CLAIMED IS:

1. A network device managing apparatus comprising:
acquiring means for specifying a device on a
network as a management target and acquiring management
5 information for the specified device;
generating means for generating output
information to display the acquired management
information in a predetermined form;
wherein if there is a link to other output
10 information in the output information, said acquiring
means acquires the management information necessary for
said other output information.
2. The apparatus according to claim 1, wherein the
management information is information in an MIB form.
- 15 3. The apparatus according to claim 1, wherein the
generating means generates the output information based
on a data which contains an HTML format for defining
the predetermined form, and a management information
item of the device.
- 20 4. The apparatus according to claim 1, wherein said
generating means generates the output information in an
HTML format.
5. The apparatus according to claim 1, wherein
further comprising output means for outputting the
25 output information.

6. The apparatus according to claim 1, wherein further comprising output means for displaying the output information in accordance with URL.

7. A network device managing method using an SNMP
5 protocol, comprising:

the generating step of specifying a device on a network as a management target and generating a command for acquiring management information for the device and setting the information in a memory;

10 the setting step of setting, on the basis of the command, management information of the device which is acquired through the network in said memory; and

the output step of outputting/displaying the set or acquired management information of the device in a
15 predetermined form.

8. The method according to claim 7, wherein the management information is information in an MIB form.

9. The method according to claim 7, wherein the command contains an HTML format for defining the
20 predetermined form, and a management information item of the device.

10. The method according to claim 7, wherein the output step comprises displaying the set or acquired result in an HTML format.

25 11. The method according to claim 7, wherein if there is an URL linked to the management information, the

setting step further comprises setting or acquiring linked management information.

12. The method according to claim 7, wherein the output step comprises displaying a result of management
5 information set or acquired in accordance with the URL.

13. A computer-readable storage medium storing a program for managing network devices using an SNMP protocol, the program comprising:

a code for the generating step of specifying a
10 device on a network as a management target and generating a command for acquiring management information for the device and setting the information in a memory;

a code for the setting step of setting, on the
15 basis of the command, management information of the device which is acquired through the network in said memory; and

a code for the output step of
outputting/displaying the set or acquired management
20 information of the device in a predetermined form.

ABSTRACT OF THE DISCLOSURE

In a network device managing method using the
SNMP protocol, a device on a network is designated as a
management target, and a command for acquiring
5 management information for the device and setting the
information in a memory is generated (S101). The
management information for the device which is acquired
through the network is set in the memory on the basis
of the command (S104, S105, S107). A set or acquired
10 result is output/displayed as management information in
a predetermined form (S106). This method makes it
possible to quickly display management information of a
device linked to the network in real time.

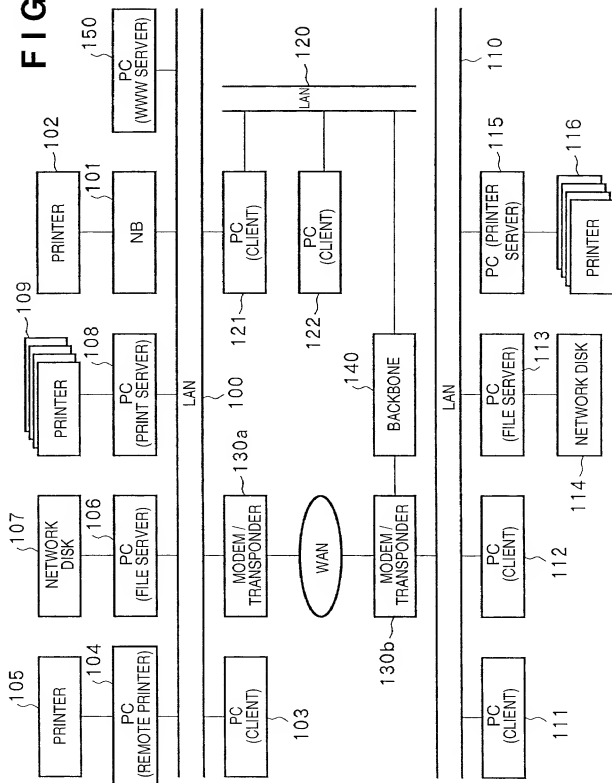
FIG. 1

FIG. 2

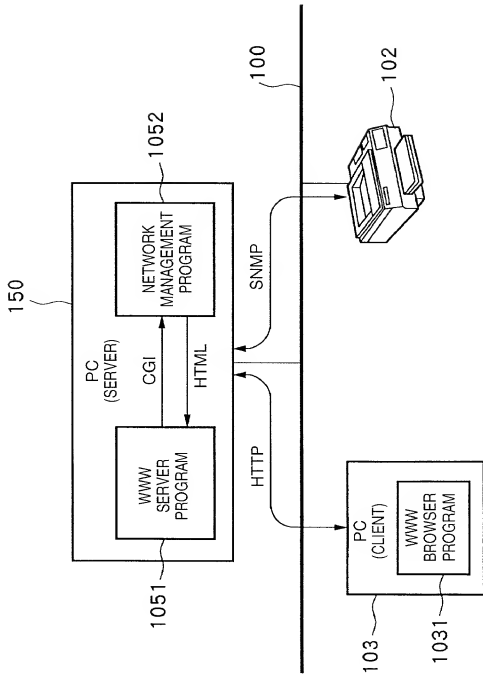
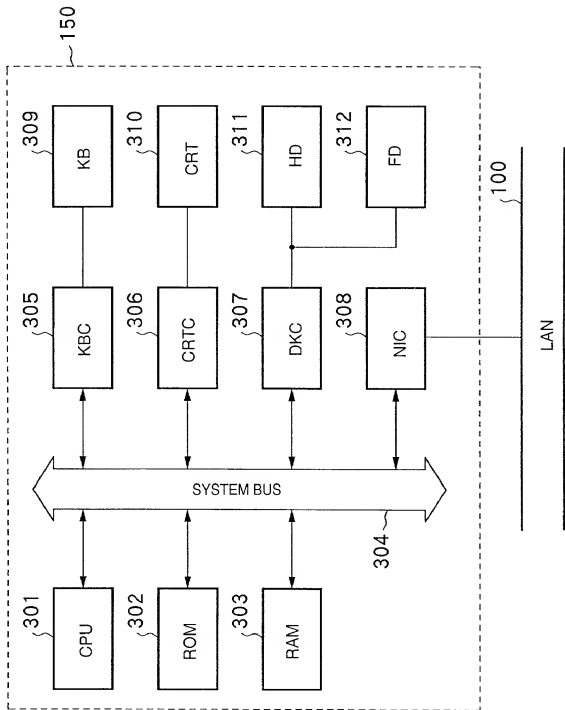
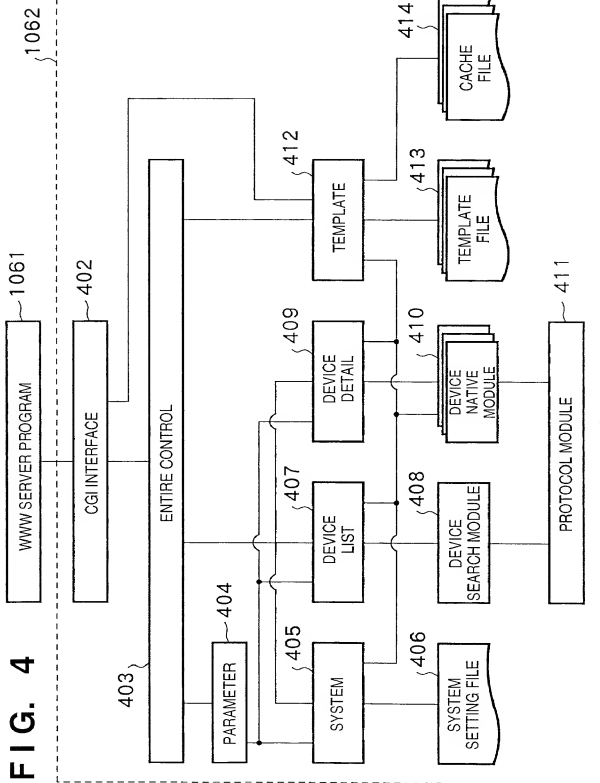


FIG. 3





```
<<TEMPLATE>>

<<HEAD>>
    DESCRIPTION OF HEAD BLOCK
<</HEAD>>

<<BODY>>
    DESCRIPTION OF BODY BLOCK
<</BODY>>

<</TEMPLATE>>
```

DESCRIPTION OF HEAD BLOCK

```
<<HEAD>>
    <<VARIABLE NAME="WNTVAR_DCV_PRODUCT">>
    <<VARIABLE NAME="WNTVAR_DCV_NETADDR">>
    <<VARIABLE NAME="WNTVAR_DCV_STATUS">>
        :
        :
<</HEAD>>
```

DESCRIPTION OF BODY BLOCK

```
<<BODY>>
    <HTML>
    <BODY>
        :
        :
    <TD><FONT SIZE="3">
    <<EMBEDDED NAME="WNTVAR_DCV_PRODUCT">>
    </FONT></TD>
        :
        :
    </BODY>
    </HTML>
<</BODY>>
```

```
<<TEMPLATE>>

<<HEAD>>
  DESCRIPTION OF HEAD BLOCK
<</HEAD>>

<<BODY>>
  DESCRIPTION OF BODY BLOCK
<</BODY>>

<</TEMPLATE>>
```

```
<<HEAD>>
<<VARIABLE NAME="WNTVAR_DCV_PRODUCT">>
<<VARIABLE NAME="WNTVAR_DCV_NETADDR">>
<<VARIABLE NAME="WNTVAR_DCV_STATUS">>
:
:
<</HEAD>>
```

```
<<BODY>>
<HTML>
<BODY>

      :
      :
<TD><FONT SIZE="3">
<<EMBEDED NAME="WNTVAR_DCV_PRODUCT">>
</FONT></TD>

      :
      :
</BODY>
</HTML>
<</BODY>>
```

FIG. 6

6/11

FILE (F) EDIT (E) VIEW (V) GO (G) FAVORITES (A) HELP (H)			
BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY CHANNEL NETWORKS MAIL FONT			
ADDRESS http://hops2/cgibin/default.html			
DEVICE NAME	PRODUCT NAME	NETWORK BOARD PRODUCT NAME	NETWORK ADDRESS STATE
A	MEDIA PS-C1	PS Network Printer Board-C1	192.168.16.134
B	GP PS/PCL-01	Network Multi-PDL Printer Board-D1	192.168.16.135
C	LASER SHOT LBP-840	JC-CONNECT C-XX00TNE	192.168.16.188
	LASER SHOT LBP-840	Ganon NB-3	192.168.16.231
D	CP660LIPS	NB-2	192.168.16.133
E	LASER SHOT LBP-930EX	NB-2	192.168.16.166
F	LASER SHOT LBP-930	NB-2	192.168.16.100
G	CP660LIPS	NB-2	192.168.16.102
H	LASER SHOT LBP-850	NB-2	192.168.16.167
I	HP LaserJet 5Si	HP ETHERNET MULTI-ENVIRONMENT. ROM A. 05. JETDIRECT. JD24. EEPROM	192.168.16.216
	HP LaserJet	HP ETHERNET MULTI-ENVIRONMENT. ROM	

FILE (F) EDIT (E) VIEW (V) GO (G) FAVORITES (A) HELP (H)

BACK FORWARD STOP REFRESH HOME SEARCH FAVORITES HISTORY

ADDRESS http://auvis-p0151/WEB81nDbg/ WNS.exe?cmd=devget&addr=192.168.16.134&tmpl=net

LINK

NETWORK MANAGEMENT SOFTWARE

RETRUN TO DEVICE LIST

STATE

INFORMATION

EQUIPMENT

ERROR

NETWORK

PROTOCOL

SYSTEM SETTING

SUPPORT LINK

MANUAL

DEVICE DETAILS

MEDIO PS-C1

LAST UPDATE : DECEMBER 4, 1998 08:57:06

STATE

PRINTER : READY

COPY : READY

FAX : READY

PAPER FEED

NAME :	PAPER FEED
MANUAL FEED	INFORMATION :
TRAY	SIZE UNKNOWN
CASSETTE 1	A4
CASSETTE 2	LETTER R
CASSETTE 3	B5
CASSETTE 4	B4
CASSETTE 5	LEGAL
CASSETTE 6	LETTER

INTRANET ZONE

FIG. 8

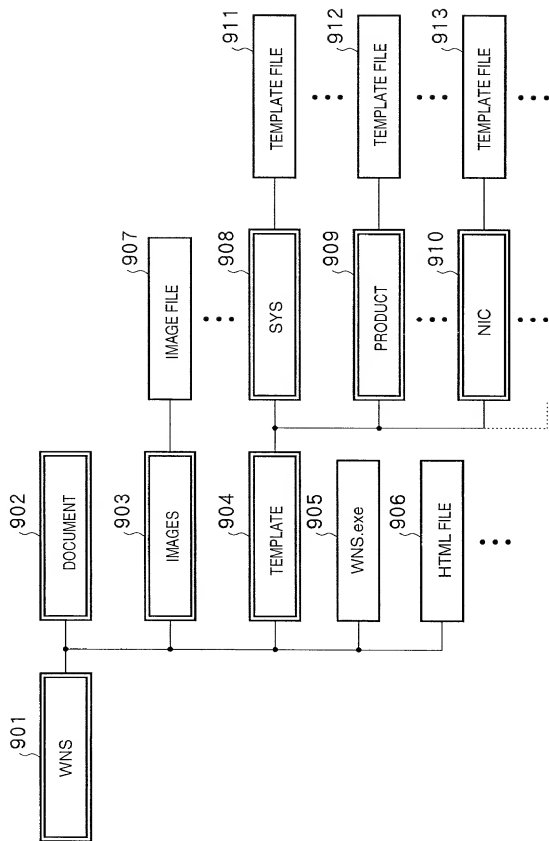


FIG. 9

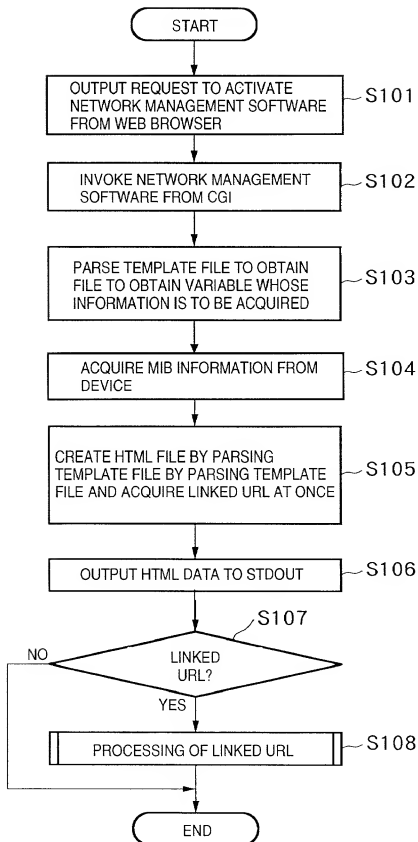


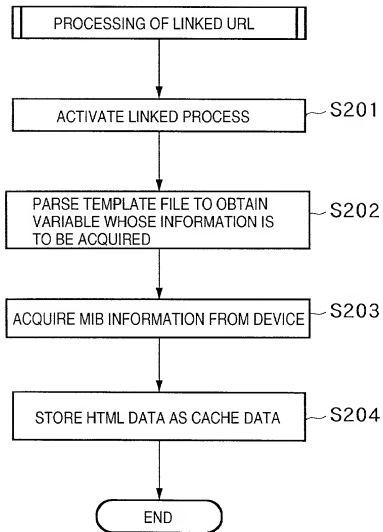
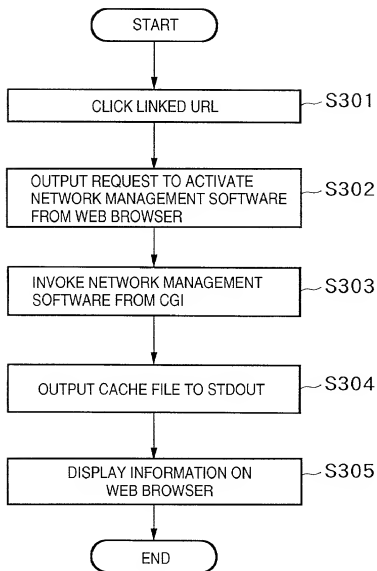
FIG. 10

FIG. 11

COMBINED DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

(Page 1)

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name:

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

NETWORK DEVICE MANAGING APPARATUS AND METHOD

AND STORAGE MEDIUM

the specification of which [x] is attached hereto. [] was filed on _____

as United States Application No. or PCT International Application No. _____

and was amended on _____ (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose information which is material to patentability as defined in 37 CFR §1.56.

I hereby claim foreign priority benefits under 35 U.S.C. §119(a)-(d) or §365(b), of any foreign application(s) for patent or inventor's certificate, or §365(a) of any PCT international application which designates at least one country other than the United States, listed below and have also identified below any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed:

<u>Country</u>	<u>Application No.</u>	<u>Filed (Day/Mo./Yr.)</u>	<u>(Yes/No)</u> <u>Priority Claimed</u>
JAPAN	11-169951	16/06/1999	Yes

I hereby appoint the practitioners associated with the firm and customer number provided below to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith, and direct that all correspondence be addressed to the address associated with that Customer Number:

FITZPATRICK, CELLA, HARPER & SCINTO **Customer Number: 05514**

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole or First Inventor Minoru TORII

Inventor's signature Minoru Torii

Date June 8, 2000 Citizen/Subject of Japan

Residence 5-6, Tamagawagakuen 1-chome, Machida-shi,

Tokyo, Japan

Post Office Address c/o CANON KABUSHIKI KAISHA,

30-2, Shimomaruko 3-chome, Ohta-ku, Tokyo, Japan